Efficient Learning of Simplices

Navin Goyal
Microsoft Research India
navingo@microsoft.com

Luis Rademacher
Computer Science and Engineering
Ohio State University
lrademac@cse.ohio-state.edu

Abstract

We show an efficient algorithm for the following problem: Given uniformly random points from an arbitrary n-dimensional simplex, estimate the simplex. The size of the sample and the number of arithmetic operations of our algorithm are polynomial in n. This answers a question of Frieze, Jerrum and Kannan [5]. Our result can also be interpreted as efficiently learning the intersection of $n+1$ half-spaces in $\mathbb{R}^n$ in the model where the intersection is bounded and we are given polynomially many uniform samples from it. Our proof uses the local search technique from Independent Component Analysis (ICA), also used by [5]. Unlike these previous algorithms, which were based on analyzing the fourth moment, ours is based on the third moment.

1 Introduction

We are given uniformly random samples from an unknown convex body in $\mathbb{R}^n$, how many samples are needed to approximately reconstruct the body? It seems intuitively clear, at least for $n = 2, 3$, that if we are given sufficiently many such samples then we can reconstruct (or learn) the body with very little error. For general n, it is known to require $2^{\Omega(\sqrt{n})}$ samples [6] (see also [11] for a similar lower bound in a different but related model of learning). This is an information-theoretic lower bound and no computational considerations are involved. As mentioned in [6], it turns out that if the body has few facets (e.g. polynomial in n), then polynomial in n samples are sufficient for approximate reconstruction. This is an information-theoretic upper bound and no efficient algorithms (i.e., with running time poly(n)) are known. (We remark that to our knowledge the same situation holds for polytopes with poly(n) vertices.) In this paper we study the reconstruction problem for the special case when the input bodies are restricted to be (full-dimensional) simplices. We show that in this case one can in fact learn the body efficiently. More precisely, the algorithm knows that the input body is a simplex but only up to an affine transformation, and the problem is to recover this affine transformation. This answers a question of Frieze et al. [5, Section 6].

The problem of learning a simplex is also closely related to the well-studied problem of learning intersections of half-spaces. Suppose that the intersection of $n+1$ half-spaces in $\mathbb{R}^n$

is bounded, and we are given poly(n) uniformly random samples from it. Then our learning simplices result directly implies that we can learn the $n+1$ half-spaces. This also has the advantage of being a proper learning algorithm, meaning that the output of the algorithm is a set of $n+1$ half-spaces, unlike many of the previous algorithms.

Previous work. Perhaps the first approach to learning simplices that comes to mind is to find a minimum volume simplex containing the samples. This can be shown to be a good approximation to the original simplex. (Such minimum volume estimators have been studied in machine learning literature, see e.g. [19] for the problem of estimating the support of a probability distribution. We are not aware of any technique that applies to our situation and provides theoretical guarantees.) However, the problem of finding a minimum volume simplex is in general NP-hard [18]. This hardness is not directly applicable for our problem because our input is a random sample and not a general point set. Nevertheless, we do not have an algorithm for directly finding a minimum volume simplex; instead we use ideas similar to those used in Independent Component Analysis (ICA). ICA studies the following model: Given a sample from an affine transformation of a random vector with independently distributed coordinates, recover the affine transformation. Frieze et al. [5] gave an efficient algorithm for this problem (with some restrictions on the allowed distributions, but also with some weaker requirements than full independence) along with most of the details of a rigorous analysis (a complete analysis of a special case can be found in Arora et al. [3]; see also Vempala and Xiao [23] for a generalization of ICA to subspaces along with a rigorous analysis). The problem of learning parallelopipeds from uniformly random samples is a special case of this problem. They [5] asked if one could learn other convex bodies, and in particular simplices, efficiently from uniformly random samples. Nguyen and Regev [17] gave a simpler and rigorous algorithm and analysis for the case of learning parallelopipeds with similarities to the popular FastICA algorithm of Hyvärinen [9]. The algorithm in [17] is a first order algorithm unlike Frieze et al.'s second order algorithm.

The algorithms in both [5, 17] make use of the fourth moment function of the probability distribution. Briefly, the fourth moment in direction $u \in \mathbb{R}^n$ is $\mathbb{E}(u \cdot X)^4$, where $X \in \mathbb{R}^n$ is the random variable distributed according to the input distribution. The moment function can be estimated from the samples. The independent components of the distribution correspond to local maxima or minima of the moment function, and can be approximately found by finding the local maxima/minima of the moment function estimated from the sample.

More information on ICA including historical remarks can be found in [10, 4]. Ideas similar to ICA have been used in statistics in the context of projection pursuit since the mid-seventies. ICA cannot be applied to the simplex learning problem directly as there is no clear independence among the components. Let us note that Frieze et al. [5] allow certain kinds of dependences among the components, however it does not appear to be useful for learning simplices.

Learning intersections of half-spaces is a well-studied problem in learning theory. The problem of PAC-learning intersections of even two half-spaces is open, and there is evidence that it is hard at least for sufficiently large number of half-spaces: E.g., Klivans and Sherstov [13] prove that learning intersections of n^ϵ half-spaces in $\mathbb{R}^n$ (for constant $\epsilon > 0$) is hard under standard cryptographic assumptions (PAC-learning is possible, however, if one also has access to a membership oracle in addition to random samples [14]). Because of this, much effort has been expended on learning when the distribution of random samples is some simple distribution, see e.g. [12, 22, 21] and references therein. This line of work makes substantial progress towards the goal of learning intersections of k half-spaces efficiently, however it falls short of being able to do this in time polynomial in k and n; in particular, these algorithms do not seem to be able to learn simplices. The distribution of samples in these works is either the Gaussian distribution or the uniform distribution over a ball. Frieze et al. [5] and Goyal and Rademacher [6] consider the uniform distribution over the intersection. Note that this requires that the intersection be bounded. Note also that one only gets positive samples in this case unlike other work on learning intersections of half-spaces. The problem of learning convex bodies can also be thought of as learning a distribution or density estimation problem for a special class of distributions.

Gravin et al. [7] show how to reconstruct a polytope with N vertices in $\mathbb{R}^n$, given its first $O(nN)$ moments in $(n+1)$ random directions. In our setting, where we have access to only a polynomial number of random samples, it's not clear how to compute moments of such high orders to the accuracy required for the algorithm of [7] even for simplices.

See [2] for a recent review of applications of the method of moments in other learning problems.

Our results For clarity of the presentation, we use the following machine model for the running time: a random access machine that allows the following exact arithmetic operations over real numbers in constant time: addition, substraction, multiplication, division and square root.

Theorem 1. *There is an algorithm (Algorithm 1 below) such that given access to random samples from a simplex $S_{INPUT} \subseteq \mathbb{R}^n$, with probability at least $1-\delta$ over the sample and the randomness of the algorithm, it outputs $n+1$ vectors that are the vertices of a simplex S so that $d_{TV}(S, S_{INPUT}) \leq \epsilon$. The algorithm runs in time polynomial in n, $1/\epsilon$ and $1/\delta$.*

As mentioned earlier, our algorithm uses ideas from ICA. Our algorithm uses the third moment instead of the fourth moment used in certain versions of ICA. The third moment is not useful for learning symmetric bodies such as the cube as it is identically 0. It is however useful for learning a simplex where it provides useful information, and is easier to handle than the fourth moment.

The probability of success of the algorithm can be "boosted" so that the dependence of the running time on δ is only linear in $\log(1/\delta)$ as follows: The following discussion uses the space of simplices with total variation distance as the underlying metric space. Let ϵ be the target distance. Take an algorithm that succeeds with probability $5/6$ and error parameter ϵ' to be fixed later (such as Algorithm 1 with $\delta = 1/6$). Run the algorithm $t = O(\log 1/\delta)$ times to get t simplices. By a Chernoff-type argument, at least $2t/3$ simplices are within ϵ' of the input simplex with probability at least $1 - \delta/2$.

By sampling, we can estimate the distances between all pairs of simplices with additive error less than $\epsilon'/10$ in time polynomial in $t, 1/\epsilon'$ and $\log 1/\delta$ so that all estimates are correct with probability at least $1-\delta/2$. For every output simplex, compute the number of output simplices within estimated distance $(2+1/10)\epsilon'$. With probability at least $1-\delta$ both of the desirable events happen, and then necessarily there is at least one output simplex, call it S, that has $2t/3$ output simplices within estimated distance $(2+1/10)\epsilon'$. Any such S must be within $(3+2/10)\epsilon'$ of the input simplex. Thus, set $\epsilon' = \epsilon/(3+2/10)$.

Idea of the algorithm. Many of the details of our proofs follow an outline similar to previous work. The main new idea is that, after putting the samples in a suitable position (see below), the third moment of the sample can be used to recover the simplex using a simple FastICA-like algorithm. We outline our algorithm next.

As any full-dimensional simplex can be mapped to any other full-dimensional simplex by an invertible affine transformation, it is enough to determine the translation and linear transformation that would take the given simplex to some canonical simplex. As is well-known for ICA-like problems (see, e.g., [5]), this transformation can be determined *up to a rotation* from the mean and the covariance matrix of the uniform distribution on the given simplex. The mean and the covariance matrix can be estimated efficiently from a sample. A convenient choice of an n-dimensional simplex is the convex hull of the canonical vectors in $\mathbb{R}^{n+1}$. We denote this simplex Δ_n and call it the *standard simplex*. So, the algorithm begins by picking an arbitrary invertible affine transformation T that maps $\mathbb{R}^n$ onto the hyperplane $\{x \in \mathbb{R}^{n+1} : \mathbb{1} \cdot x = 1\}$. We use a T so that $T^{-1}(\Delta_n)$ is an isotropic[1] simplex. In this case, the algorithm brings the sample set into isotropic position and embeds it in $\mathbb{R}^{n+1}$ using T. After applying these transformations we may assume (at the cost of small errors in the final result) that our sample set is obtained by sampling from an unknown rotation of the standard simplex that leaves the all-ones vector (denoted $\mathbb{1}$ from now on) invariant (thus this rotation keeps the center of mass of the standard simplex fixed), and the problem is to recover this rotation.

To find the rotation, the algorithm will find the vertices of the rotated simplex approximately. This can be done efficiently because of the following characterization of the vertices: Project the vertices of the simplex onto the hyperplane through the origin orthogonal to $\mathbb{1}$ and normalize the resulting vectors. Let V denote this set of $n+1$ points. Consider the problem of maximizing the 3rd moment of the uniform distribution in the simplex along unit vectors orthogonal to $\mathbb{1}$. Then V is the complete set of local maxima and the complete set of global maxima (Theorem 6). A fixed point-like iteration (inspired by the analysis of gradient descent in [17]) starting from a random point in the unit sphere finds a local maximum efficiently with high probability. By the analysis of the coupon collector's problem, $O(n \log n)$ repetitions are highly likely to find all local maxima.

Idea of the analysis. In the analysis, we first argue that after putting the sample in isotropic position and mapping it through T, it is enough to analyze the algorithm in the

[1]See Section 2.

case where the sample comes from a simplex S that is close to a simplex S' that is the result of applying a rotation leaving $\mathbb{1}$ invariant to the standard simplex. The closeness here depends on the accuracy of the sample covariance and mean as an estimate of the input simplex's covariance matrix and mean. A sample of size $O(n)$ guarantees ([1, Theorem 4.1], [20, Corollary 1.2]) that the covariance and mean are close enough so that the uniform distributions on S and S' are close in total variation. We show that the subroutine that finds the vertices (Subroutine 1), succeeds with some probability when given a sample from S'. By definition of total variation distance, Subroutine 1 succeeds with almost as large probability when given a sample from S (an argument already used in [17]). As an additional simplifying assumption, it is enough to analyze the algorithm (Algorithm 1) in the case where the input is isotropic, as the output distribution of the algorithm is equivariant with respect to affine invertible transformations as a function of the input distribution.

Organization of the paper. Starting with some preliminaries in Sec. 2, we state some results on the third moment of simplices in Sec. 3. In Sec. 4 we give an algorithm that estimates individual vertices of simplices in a special position; using this algorithm as a subroutine in Sec. 5 we give the algorithm for the general case. Sec. 6 characterizes the set of local maxima of the third moment.

2 Preliminaries

An n-simplex is the convex hull of $n+1$ points in $\mathbb{R}^n$. We will be interested in the non-degenerate case where these $n+1$ points do not lie on an $(n-1)$-dimensional affine hyperplane. It will be convenient to work with the standard n-simplex Δ^n living in $\mathbb{R}^{n+1}$ defined as the convex hull of the $n+1$ canonical unit vectors $e_1, \dots, e_{n+1}$; that is

$$\Delta^n = \{(x_0, \dots, x_n) \in \mathbb{R}^{n+1} \,|x_0 + \cdots + x_n = 1 \\ \text{and } x_i \geq 0 \text{ for all } i\}.$$

The canonical simplex Ω^n living in $\mathbb{R}^n$ is given by

$$\{(x_0, \dots, x_{n-1}) \in \mathbb{R}^n \,|x_0 + \cdots + x_{n-1} \leq 1 \\ \text{and } x_i \geq 0 \text{ for all } i\}.$$

Note that Δ^n is the facet of Ω^{n+1} opposite to the origin.

Let B_n denote the n-dimensional Euclidean ball.

The complete homogeneous symmetric polynomial of degree d in variables $u_0, \dots, u_n$, denoted $h_n(u_0, \dots, u_n)$, is the sum of all monomials of degree d in the variables:

$$\begin{aligned} h_d(u_0, \dots, u_n) &= \sum_{k_0 + \cdots + k_n = d} u_0^{k_0} \cdots u_n^{k_n} \\ &= \sum_{0 \leq i_0 \leq i_1 \leq \cdots \leq i_d \leq n} u_{i_0} u_{i_1} \cdots u_{i_d}. \end{aligned}$$

Also define the d-th power sum as

$$p_d(u_0, \ldots, u_n) = u_0^d + \ldots + u_n^d.$$

For a vector $u = (u_0, u_1, \ldots, u_n)$, we define

$$u^{(2)} = (u_0^2, u_1^2, \ldots, u_n^2).$$

Vector $\mathbb{1}$ denotes the all ones vector (the dimension of the vector will be clear from the context).

A random vector $X \in \mathbb{R}^n$ is *isotropic* if $\mathbb{E}(X) = 0$ and $\mathbb{E}(XX^T) = I$. A compact set in $\mathbb{R}^n$ is isotropic if a uniformly distributed random vector in it is isotropic. The inradius of an isotropic n-simplex is $\sqrt{(n+2)/n}$, the circumradius is $\sqrt{n(n+2)}$.

The total variation distance between two probability measures is $d_{TV}(\mu, \nu) = \sup_A |\mu(A) - \nu(A)|$ for measurable A. For two compact sets $K, L \subseteq \mathbb{R}^n$, we define the total variation distance $d_{TV}(K, L)$ as the total variation distance between the corresponding uniform distributions on each set. It can be expressed as

$$d_{TV}(K, L) = \begin{cases} \frac{\operatorname{vol} K \setminus L}{\operatorname{vol} K} & \text{if } \operatorname{vol} K \geq \operatorname{vol} L, \\ \frac{\operatorname{vol} L \setminus K}{\operatorname{vol} L} & \text{if } \operatorname{vol} L > \operatorname{vol} K. \end{cases}$$

This identity implies the following elementary estimate:

Lemma 2. *Let K, L be two compact sets in $\mathbb{R}^n$. Let $0 < \alpha \leq 1 \leq \beta$ such that $\alpha K \subseteq L \subseteq \beta K$. Then $d_{TV}(K, L) \leq 2\left(1 - (\alpha/\beta)^n\right)$.*

Proof. We have $d_{TV}(\alpha K, \beta K) = 1 - (\alpha/\beta)^n$. Triangle inequality implies the desired inequality. □

Lemma 3. *Consider the coupon collector's problem with n coupons where every coupon occurs with probability at least α. Let $\delta > 0$. Then with probability at least $1 - \delta$ all coupons are collected after $\alpha^{-1}(\log n + \log 1/\delta)$ trials.*

Proof. The probability that a particular coupon is not collected after that many trials is at most

$$(1-\alpha)^{\alpha^{-1}(\log n + \log 1/\delta)} \leq e^{-\log n - \log 1/\delta} = \delta/n.$$

The union bound over all coupons implies the claim. □

3 Computing the moments of a simplex

The k-th moment $m_k(u)$ over Δ^n is the function

$$u \mapsto \mathbb{E}_{X \in \Delta_n}((u \cdot X)^k).$$

In this section we present a formula for the moment over Δ^n. Similar more general formulas appear in [15]. We will use the following result from [8] for $\alpha_i \geq 0$:

$$\int_{\Omega^{n+1}} x_0^{\alpha_0} \cdots x_n^{\alpha_n} dx = \frac{\alpha_0! \cdots \alpha_n!}{(n+1+\sum_i \alpha_i)!}.$$

From the above we can easily derive a formula for integration over Δ^n:

$$\int_{\Delta^n} x_0^{\alpha_0} \cdots x_n^{\alpha_n} dx = \sqrt{n+1} \cdot \frac{\alpha_0! \cdots \alpha_n!}{(n+\sum_i \alpha_i)!}.$$

Now

$$\begin{aligned}
&\int_{\Delta^n} (x_0 u_0 + \ldots + x_n u_n)^k dx \\
&= \sum_{k_0+\cdots+k_n=k} \binom{k}{k_0!, \ldots, k_n!} u_0^{k_0} \ldots u_n^{k_n} \int_{\Delta^n} x_0^{k_0} \ldots x_n^{k_n} dx \\
&= \sum_{k_0+\cdots+k_n=k} \binom{k}{k_0!, \ldots, k_n!} u_0^{k_0} \ldots u_n^{k_n} \frac{\sqrt{n+1} \cdot k_0! \ldots k_n!}{(n+\sum_i k_i)!} \\
&= \frac{k!\sqrt{n+1}}{(n+k)!} \sum_{k_0+\cdots+k_n=k} u_0^{k_0} \ldots u_n^{k_n} \\
&= \frac{k!\sqrt{n+1}}{(n+k)!} h_k(u).
\end{aligned}$$

The variant of Newton's identities for the complete homogeneous symmetric polynomial gives the following relations which can also be verified easily by direct computation:

$$3h_3(u) = h_2(u)p_1(u) + h_1(u)p_2(u) + p_3(u),$$

$$2h_2(u) = h_1(u)p_1(u) + p_2(u) = p_1(u)^2 + p_2(u).$$

Divide the above integral by the volume of the standard simplex $|\Delta_n| = \sqrt{n+1}/n!$ to get the moment:

$$\begin{aligned}
m_3(u) &= \frac{3!\sqrt{n+1}}{(n+3)!} h_3(u)/|\Delta_n| \\
&= \frac{2(h_2(u)p_1(u) + h_1(u)p_2(u) + p_3(u))}{(n+1)(n+2)(n+3)} \\
&= \frac{(p_1(u)^3 + 3p_1(u)p_2(u) + 2p_3(u))}{(n+1)(n+2)(n+3)}.
\end{aligned}$$

4 Subroutine for finding the vertices of a rotated standard simplex

In this section we solve the following simpler problem: Suppose we have poly(n) samples from a rotated copy S of the standard simplex, where the rotation is such that it leaves $\mathbb{1}$ invariant. The problem is to approximately estimate the vertices of the rotated simplex from the samples.

We will analyze our algorithm in the coordinate system in which the input simplex is the standard simplex. This is only for convenience in the analysis and the algorithm itself does not know this coordinate system.

As we noted in the introduction, our algorithm is inspired by the algorithm of Nguyen and Regev [17] for the related problem of learning hypercubes and also by the FastICA algorithm [9]. New ideas are needed for our algorithm for learning simplices; in particular, our update rule is different. With the right update rule in hand the analysis turns out to be quite similar to the one in [17].

We want to find local maxima of the sample third moment. A natural approach to do this would be to use gradient descent or Newton's method (this was done in [5]). Our algorithm, which only uses first order information, can be thought of as a fixed point algorithm leading to a particularly simple analysis and fast convergence. Before stating our algorithm we describe the update rule we use.

We will use the abbreviation $C_n = (n+1)(n+2)(n+3)/6$. Then, from the expression for $m_3(u)$ we get

$$\nabla m_3(u) = \frac{1}{6C_n}\left(3p_1(u)^2\mathbb{1} + 3p_2(u)\mathbb{1} + 6p_1(u)u + 6u^{(2)}\right).$$

Solving for $u^{(2)}$ we get

$$\begin{aligned} u^{(2)} &= C_n\nabla m_3(u) - \frac{1}{2}p_1(u)^2\mathbb{1} - \frac{1}{2}p_2(u)\mathbb{1} - p_1(u)u \\ &= C_n\nabla m_3(u) - \frac{1}{2}(u\cdot\mathbb{1})^2\mathbb{1} - \frac{1}{2}(u\cdot u)^2\mathbb{1} - (u\cdot\mathbb{1})u. \qquad (1) \end{aligned}$$

While the above expressions are in the coordinate system where the input simplex is the canonical simplex, the important point is that all terms in the last expression can be computed in any coordinate system that is obtained by a rotation leaving $\mathbb{1}$ invariant. Thus, we can compute $u^{(2)}$ as well independently of what coordinate system we are working in. This immediately gives us the algorithm below. We denote by $\hat{m}_3(u)$ the sample third moment, i.e., $\hat{m}_3(u) = \frac{1}{t}\sum_{i=1}^{t}(u\cdot r_i)^3$ for t samples. This is a polynomial in u, and the gradient is computed in the obvious way. Moreover, the gradient of the sample moment is clearly an unbiased estimator of the gradient of the moment; a bound on the deviation is given in the analysis (Lemma 4). For each evaluation of the gradient of the sample moment, we use a fresh sample.

It may seem a bit alarming that the fixed point-like iteration is squaring the coordinates of u, leading to an extremely fast growth (see Equation 1 and Subroutine 1). But, as in other algorithms having quadratic convergence like certain versions of Newton's method, the convergence is very fast and the number of iterations is small. We show below that it is $O(\log(n/\delta))$, leading to a growth of u that is polynomial in n and $1/\delta$. The boosting argument described in the introduction makes the final overall dependence in δ to be only linear in $\log(1/\delta)$.

We state the following subroutine for $\mathbb{R}^n$ instead of $\mathbb{R}^{n+1}$ (thus it is learning a rotated copy of Δ^{n-1} instead of Δ^n). This is for notational convenience so that we work with n instead of $n+1$.

Subroutine 1 Find one vertex of a rotation of the standard simplex Δ^{n-1} via a fixed point iteration-like algorithm

```
Input: Samples from a rotated copy of the $n$-dimensional standard simplex (for a rotation that leaves $\mathbb{1}$ invariant).
Output: An approximation to a uniformly random vertex of the input simplex.

Pick $u(1) \in S^{n-1}$, uniformly at random.
**for** $i = 1$ to $r$ **do**
    $u(i+1) := C_{n-1} \nabla \hat{m}_3(u(i)) - \frac{1}{2}(u(i)\cdot\mathbb{1})^2\mathbb{1} - \frac{1}{2}(u(i)\cdot u(i))^2\mathbb{1} - (u(i)\cdot\mathbb{1})u(i).$
    Normalize $u(i+1)$ by dividing by $\|u(i+1)\|_2$.
**end for**
Output $u(r+1)$.
```

Lemma 4. *Let $c > 0$ be a constant, $n > 20$, and $0 < \delta < 1$. Suppose that Subroutine 1 uses a sample of size $t = 2^{17} n^{2c+22} (\frac{1}{\delta})^2 \ln \frac{2n^5 r}{\delta}$ for each evaluation of the gradient and runs for $r = \log \frac{4(c+3)n^2 \ln n}{\delta}$ iterations. Then with probability at least $1-\delta$ Subroutine 1 outputs a vector within distance $1/n^c$ from a vertex of the input simplex. With respect of the process of picking a sample and running the algorithm, each vertex is equally likely to be the nearest.*

Note that if we condition on the sample, different vertices are not equally likely over the randomness of the algorithm. That is, if we try to find all vertices running the algorithm multiple times on a fixed sample, different vertices will be found with different likelihoods.

Proof. Our analysis has the same outline as that of Nguyen and Regev [17]. This is because the iteration that we get is the same as that of [17] except that cubing is replaced by squaring (see below); however some details in our proof are different. In the proof below, several of the inequalities are quite loose and are so chosen to make the computations simpler.

We first prove the lemma assuming that the gradient computations are exact and then show how to handle samples. We will carry out the analysis in the coordinate system where the given simplex is the standard simplex. This is only for the purpose of the analysis, and this coordinate system is not known to the algorithm. Clearly, $u(i+1) = (u(i)_1^2, \dots, u(i)_n^2)$. It follows that,

$$u(i+1) = (u(1)_1^{2^i}, \dots, u(1)_n^{2^i}).$$

Now, since we choose $u(1)$ randomly, with probability at least $(1 - (n^2 - n)\delta')$ one of the coordinates of $u(1)$ is greater than all the other coordinates in absolute value by a factor of at least $(1+\delta')$, where $0 < \delta' < 1$. (A similar argument is made in [17] with different parameters. We briefly indicate the proof for our case: The probability that the event in question does not happen is less than the probability that there are two coordinates $u(1)_a$ and $u(1)_b$ such that their absolute values are within factor $1 + \delta'$, i.e. $1/(1+\delta') \le |u(1)_a|/|u(1)_b| < 1 + \delta'$. The probability that for given a, b this event happens can be seen as the Gaussian area of the four sectors (corresponding to the four choices of signs of $u(1)_a, u(1)_b$) in the plane each with angle less than $2\delta'$. By symmetry, the Gaussian volume of these sectors is $2\delta'/(\pi/2) < 2\delta'$. The probability that such a pair (a, b) exists is less than $2\binom{n}{2}\delta'$.) Assuming this happens, then after r iterations, the ratio between the largest coordinate (in absolute value) and the absolute value of any other coordinate is at least $(1+\delta')^{2^r}$. Thus, one of the coordinates is very close to 1 and others are very close to 0, and so $u(r+1)$ is very close to a vertex of the input simplex.

Now we drop the assumption that the gradient is known exactly. For each evaluation of the gradient we use a fresh subset of samples of t points. Here t is chosen so that each evaluation of the gradient is within ℓ_2-distance $1/n^{c_1}$ from its true value with probability at least $1 - \delta''$, where c_1 will be set at the end of the proof. An application of the Chernoff bound yields that we can take $t = 200n^{2c_1+4} \ln \frac{2n^3}{\delta''}$; we omit the details. Thus all the r evaluations of the gradient are within distance $1/n^{c_1}$ from their true values with probability at least $1 - r\delta''$.

We assumed that our starting vector $u(1)$ has a coordinate greater than every other coordinate by a factor of $(1+\delta')$ in absolute value; let us assume without loss of generality that this is the first coordinate. Hence $|u(1)_1| \ge 1/\sqrt{n}$. When expressing $u^{(2)}$ in terms of the gradient, the gradient gets multiplied by $C_{n-1} < n^3$ (we are assuming $n > 20$), keeping this in mind and letting $c_2 = c_1 - 3$ we get for $j \ne 1$

$$\frac{|u(i+1)_1|}{|u(i+1)_j|} \ge \frac{u(i)_1^2 - 1/n^{c_2}}{u(i)_j^2 + 1/n^{c_2}} \ge \frac{u(i)_1^2(1 - n^{-(c_2-1)})}{u(i)_j^2 + 1/n^{c_2}}.$$

If $u(i)_j^2 > 1/n^{c_2 - c_3}$, where $1 \le c_3 \le c_2 - 2$ will be determined later, then we get

$$\begin{aligned} |u(i+1)_1|/|u(i+1)_j| &> \frac{1 - 1/n^{c_2-1}}{1 + 1/n^{c_3}} \cdot \left(\frac{u(i)_1}{u(i)_j}\right)^2 \\ &> (1 - 1/n^{c_3})^2 \left(\frac{u(i)_1}{u(i)_j}\right)^2. \end{aligned} \tag{2}$$

Else,

$$\begin{aligned}|u(i+1)_1|/|u(i+1)_j| &> \frac{1/n - 1/n^{c_2}}{1/n^{c_2-c_3} + 1/n^{c_2}} \\ &> \left(1 - \frac{1}{n^{c_3}}\right)^2 \cdot n^{c_2-c_3-1} \\ &> \frac{1}{2} n^{c_2-c_3-1},\end{aligned}$$

where we used $c_3 \geq 1$ and $n > 20$ in the last inequality.

We choose c_3 so that

$$\left(1 - \frac{1}{n^{c_3}}\right)^2 (1+\delta') > (1+\delta'/2). \tag{3}$$

For this, $\delta' \geq 32/n^{c_3}$ or equivalently $c_3 \geq (\ln(32/\delta'))/\ln n$ suffices.

For c_3 satisfying (3) we have $(1 - \frac{1}{n^{c_3}})^2(1+\delta')^2 > (1+\delta')$. It then follows from (2) that the first coordinate continues to remain the largest in absolute value by a factor of at least $(1+\delta')$ after each iteration. Also, once we have $|u(i)_1|/|u(i)_j| > \frac{1}{2}n^{c_2-c_3-1}$, we have $|u(i')_1|/|u(i')_j| > \frac{1}{2}n^{c_2-c_3-1}$ for all $i' > i$.

(2) gives that after r iterations we have

$$\begin{aligned}\frac{|u(r+1)_1|}{|u(r+1)_j|} &> (1 - 1/n^{c_3})^{2+2^2+\ldots+2^r} \left(\frac{u(1)_1}{u(1)_j}\right)^{2^r} \\ &\geq (1 - 1/n^{c_3})^{2^{r+1}-2}(1+\delta')^{2^r}.\end{aligned}$$

Now if r is such that $(1 - 1/n^{c_3})^{2^{r+1}-2}(1+\delta')^{2^r} > \frac{1}{2}n^{c_2-c_3-1}$, we will be guaranteed that $|u(r+1)_1|/|u(r+1)_j| > \frac{1}{2}n^{c_2-c_3-1}$. This condition is satisfied if we have $(1 - 1/n^{c_3})^{2^{r+1}}(1+\delta')^{2^r} > \frac{1}{2}n^{c_2-c_3-1}$, or equivalently $((1-1/n^{c_3})^2(1+\delta'))^{2^r} \geq \frac{1}{2}n^{c_2-c_3-1}$. Now using (3) it suffices to choose r so that $(1+\delta'/2)^{2^r} \geq \frac{1}{2}n^{c_2-c_3-1}$. Thus we can take $r = \log(4(c_2-c_3)(\ln n)/\delta')$.

Hence we get $|u(r+1)_1|/|u(r+1)_j| > \frac{1}{2}n^{c_2-c_3-1}$. It follows that for $u(r+1)$, the ℓ_2-distance from the vertex $(1, 0, \ldots, 0)$ is at most $8/n^{c_2-c_3-2} < 1/n^{c_2-c_3-3}$ for $n > 20$; we omit easy details.

Now we set our parameters: $c_3 = 1 + (\ln(32/\delta')/\ln n)$ and $c_2 - c_3 - 3 = c$ and $c_1 = c_2 + 3 = 7 + c + \ln(32/\delta')/\ln n$ satisfies all the constraints we imposed on c_1, c_2, c_3. Choosing $\delta'' = \delta'/r$, we get that the procedure succeeds with probability at least $1 - (n^2 - n)\delta' - r\delta'' > 1 - n^2\delta'$. Now setting $\delta' = \delta/n^2$ gives the overall probability of error δ, and the number of samples and iterations as claimed in the lemma. □

5 Learning simplices

In this section we give our algorithm for learning general simplices, which uses the algorithm from the previous section as a subroutine. The learning algorithm uses an affine map T :

$\mathbb{R}^n \to \mathbb{R}^{n+1}$ that maps some isotropic simplex to the standard simplex. We describe now a way of constructing such a map: Let A be a matrix having as columns an orthonormal basis of $\mathbb{1}^\perp$ in $\mathbb{R}^{n+1}$. To compute one such A, one can start with the $(n+1)$-by-$(n+1)$ matrix B that has ones in the diagonal and first column, everything else is zero. Let $QR = B$ be a QR-decomposition of B. By definition we have that the first column of Q is parallel to $\mathbb{1}$ and the rest of the columns span $\mathbb{1}^\perp$. Given this, let A be the matrix formed by all columns of Q except the first. We have that the set $\{A^T e_i\}$ is the set of vertices of a regular n-simplex. Each vertex is at distance

$$\sqrt{\left(1 - \frac{1}{n+1}\right)^2 + \frac{n}{(n+1)^2}} = \sqrt{\frac{n}{n+1}}$$

from the origin, while an isotropic simplex has vertices at distance $\sqrt{n(n+2)}$ from the origin. So an affine transformation that maps an isotropic simplex in $\mathbb{R}^n$ to the standard simplex in $\mathbb{R}^{n+1}$ is $T(x) = \frac{1}{\sqrt{(n+1)(n+2)}} Ax + \frac{1}{n+1}\mathbb{1}_{n+1}$.

To simplify the analysis, we pick a new sample $r(1), \dots, r(t_3)$ to find every vertex, as this makes every vertex equally likely to be found when given a sample from an isotropic simplex. (The core of the analysis is done for an isotropic simplex; this is enough as the algorithm's first step is to find an affine transformation that puts the input simplex in approximately isotropic position. The fact that this approximation is close in total variation distance implies that it is enough to analyze the algorithm for the case of exact isotropic position, the analysis carries over to the approximate case with a small loss in the probability of success. See the proof below for the details.) A practical implementation may prefer to select one such sample outside of the for loop, and find all the vertices with just that sample—an analysis of this version would involve bounding the probability that each vertex is found (given the sample, over the choice of the starting point of gradient descent) and a variation of the coupon collector's problem with coupons that are not equally likely.

Proof of Theorem 1. As a function of the input simplex, the distribution of the output of the algorithm is equivariant under invertible affine transformations. Namely, if we apply an affine transformation to the input simplex, the distribution of the output is equally transformed.[2] The notion of error, total variation distance, is also invariant under invertible affine transformations. Therefore, it is enough to analyze the algorithm when the input simplex is in isotropic position. In this case $\|p(i)\| \leq n + 1$ (see Section 2) and we can set $t_1 \leq \operatorname{poly}(n, 1/\epsilon', \log(1/\delta))$ so that $\|\mu\| \leq \epsilon'$ with probability at least $1 - \delta/10$ (by an easy application of Chernoff's bound), for some ϵ' to be fixed later. Similarly, using results from [1, Theorem 4.1], a choice of $t_1 \leq n\epsilon'^{-2} \operatorname{polylog}(1/\epsilon') \operatorname{polylog}(1/\delta)$ implies that the empirical

[2]To see this: the equivariance of the algorithm as a map between distributions is implied by the equivariance of the algorithm on any given input sample. Now, given the input sample, if we apply an affine transformation to it, this transformation is undone except possibly for a rotation by the step $s(i) = B^{-1}(r(i) - \mu)$. A rotation may remain because of the ambiguity in the characterization of B. But the steps of the algorithm that follow the definition of $s(i)$ are equivariant under rotation, and the ambiguous rotation will be removed at the end when B is applied again in the last step.

Algorithm 1 Learning a simplex.

```
Input: Error parameter $\epsilon > 0$. Probability of failure parameter $\delta > 0$. Oracle access to random points from some $n$-dimensional simplex $S_{INPUT}$.
Output: $V = \{v(1), \dots, v(n+1)\} \subseteq \mathbb{R}^n$ (approximations to the vertices of the simplex).
Estimate the mean and covariance using $t_1 = \text{poly}(n, 1/\epsilon, 1/\delta)$ samples $p(1), \dots, p(t_1)$:
    $\mu = \frac{1}{t_1} \sum_i p(i)$,
    $\Sigma = \frac{1}{t_1} \sum_i (p(i) - \mu)(p(i) - \mu)^T$.
Compute a matrix $B$ so that $\Sigma = BB^T$ (say, Cholesky decomposition).
Let $U = \emptyset$.
for $i = 1$ to $m$ (with $m = \text{poly}(n, \log 1/\delta)$) do
    Get $t_3 = \text{poly}(n, 1/\epsilon, \log 1/\delta)$ samples $r(1), \dots r(t_3)$ and use $\mu, B$ to map them to samples $s(i)$ from a nearly-isotropic simplex: $s(i) = B^{-1}(r(i) - \mu)$.
    Embed the resulting samples in $\mathbb{R}^{n+1}$ as a sample from an approximately rotated standard simplex: Let $l(i) = T(s(i))$.
    Invoke Subroutine 1 with sample $l(1), \dots, l(t_3)$ to get $u \in \mathbb{R}^{n+1}$.
    Let $\tilde{u}$ be the nearest point to $u$ in the affine hyperplane $\{x : x \cdot \mathbb{1} = 1\}$. If $\tilde{u}$ is not within $1/\sqrt{2}$ of a point in $U$, add $\tilde{u}$ to $U$. (Here $1/\sqrt{2}$ is half of the edge length of the standard simplex.)
end for
Let
    $V = BT^{-1}(U) + \mu$
    $\phantom{V} = \sqrt{(n+1)(n+2)}BA^T \left(U - \frac{1}{n+1}\mathbb{1}\right) + \mu$.
```

second moment matrix

$$\bar{\Sigma} = \frac{1}{t_1} \sum_i p(i) p(i)^T$$

satisfies $\|\bar{\Sigma} - I\| \leq \epsilon'$ with probability at least $1 - \delta/10$. We have $\Sigma = \bar{\Sigma} - \mu\mu^T$ and this implies $\|\Sigma - I\| \leq \|\bar{\Sigma} - I\| + \|\mu\mu^T\| \leq 2\epsilon'$. Now, $s(1), \dots, s(t_3)$ is an iid sample from a simplex $S' = B^{-1}(S_{INPUT} - \mu)$. Simplex S' is close in total variation distance to some isotropic simplex[3] S_{ISO}. More precisely, Lemma 5 below shows that

$$d_{TV}(S', S_{ISO}) \leq 12n\epsilon', \tag{4}$$

with probability at least $1 - \delta/5$.

Assume for a moment that $s(1), \dots, s(t_3)$ are from S_{ISO}. The analysis of Subroutine 1 (fixed point-like iteration) given in Lemma 4 would guarantee the following: Successive invocations to Subroutine 1 find approximations to vertices of $T(S_{ISO})$ within Euclidean distance ϵ'' for some ϵ'' to be determined later and $t_3 = \operatorname{poly}(n, 1/\epsilon'', \log 1/\delta)$. We ask for each invocation to succeed with probability at least $1-\delta/(20m)$ with $m = n(\log n + \log 20/\delta)$. Note that each vertex is equally likely to be found. The choice of m is so that, if all m invocations succeed (which happens with probability at least $1 - \delta/20$), then the analysis of the coupon collector's problem, Lemma 3, implies that we fail to find a vertex with probability at most $\delta/20$. Overall, we find all vertices with probability at least $1 - \delta/10$.

But in reality samples $s(1), \dots, s(t_3)$ are from S', which is only *close* to S_{ISO}. The estimate from (4) with appropriate $\epsilon' = \operatorname{poly}(1/n, \epsilon'', \delta)$ gives

$$d_{TV}(S', S_{ISO}) \leq \frac{\delta}{10} \frac{1}{t_3 m},$$

which implies that the total variation distance between the joint distribution of all $t_3 m$ samples used in the loop and the joint distribution of actual samples from the isotropic simplex S_{ISO} is at most $\delta/10$, and this implies that the loop finds approximations to all vertices of $T(S_{ISO})$ when given samples from S' with probability at least $1-\delta/5$. The points in U are still within Euclidean distance ϵ'' of corresponding vertices of $T(S_{ISO})$.

To conclude, we turn our estimate of distances between estimated and true vertices into a total variation estimate, and map it back to the input simplex. Let $S'' = \operatorname{conv} T^{-1}U$. As T maps an isotropic simplex to a standard simplex, we have that $\sqrt{(n+1)(n+2)}T$ is an isometry, and therefore the vertices of S'' are within distance $\epsilon''/\sqrt{(n+1)(n+2)}$ of the corresponding vertices of S_{ISO}. Thus, the corresponding support functions are uniformly within

$$\epsilon''' = \epsilon''/\sqrt{(n+1)(n+2)}$$

of each other on the unit sphere. This and the fact that $S_{ISO} \supseteq B_n$ imply

$$(1 - \epsilon''')S_{ISO} \subseteq S'' \subseteq (1 + \epsilon''')S_{ISO}.$$

[3]The isotropic simplex S_{ISO} will typically be far from the (isotropic) input simplex, because of the ambiguity up to orthogonal transformations in the characterization of B.

Thus, by Lemma 2, $d_{TV}(S'', S_{ISO}) \le 1 - (\frac{1-\epsilon'''}{1+\epsilon'''})^n \le 1 - (1-\epsilon''')^{2n} \le 2n\epsilon''' \le 2\epsilon''$ and this implies that the total variation distance between the uniform distributions on $\operatorname{conv} V$ and the input simplex is at most $2\epsilon''$. Over all random choices, this happens with probability at least $1 - 2\delta/5$. We set $\epsilon'' = \epsilon/2$. ☐

Lemma 5. *Let S_{INPUT} be an n-dimensional isotropic simplex. Let Σ be an n-by-n positive definite matrix such that $\|\Sigma - I\| \le \epsilon < 1/2$. Let μ be an n-dimensional vector such that $\|\mu\| \le \epsilon$. Let B be an n-by-n matrix such that $\Sigma = BB^T$. Let S be the simplex $B^{-1}(S_{INPUT} - \mu)$. Then there exists an isotropic simplex S_{ISO} such that $d_{TV}(S, S_{ISO}) \le 6n\epsilon$.*

Proof. We use an argument along the lines of the orthogonal Procrustes problem (nearest orthogonal matrix to B^{-1}, already in [17, Proof of Theorem 4]): Let UDV^T be the singular value decomposition of B^{-1}. Let $R = UV^T$ be an orthogonal matrix (that approximates B^{-1}). Let $S_{ISO} = RS_{INPUT}$.

We have $S = UDV^T(S_{INPUT} - \mu)$. Let σ_{min}, σ_{max} be the minimum and maximum singular values of D, respectively. This implies:

$$\begin{gathered}\sigma_{min} UV^T(S_{INPUT} - \mu) \subseteq S \subseteq \sigma_{max} UV^T(S_{INPUT} - \mu),\\ \sigma_{min}(S_{ISO} - R\mu) \subseteq S \subseteq \sigma_{max}(S_{ISO} - R\mu). \end{gathered} \tag{5}$$

As $S_{ISO} \supseteq B_n$, $\|\mu\| \le 1$, R is orthogonal and S_{ISO} is convex, we have

$$S_{ISO} - R\mu \supseteq (1 - \|\mu\|)S_{ISO}.$$

Also,

$$\begin{aligned} S_{ISO} - R\mu &\subseteq S_{ISO} + \|\mu\| B_n \\ &\subseteq S_{ISO}(1 + \|\mu\|). \end{aligned}$$

This in (5) gives

$$\sigma_{min}(1 - \|\mu\|)S_{ISO} \subseteq S \subseteq \sigma_{max}(1 + \|\mu\|)S_{ISO}.$$

This and Lemma 2 imply

$$d_{TV}(S, S_{ISO}) \le 2\left(1 - \left(\frac{\sigma_{min}(1 - \|\mu\|)}{\sigma_{max}(1 + \|\mu\|)}\right)^n\right).$$

The estimate on Σ gives $\sigma_{min} \ge \sqrt{1-\epsilon}$, $\sigma_{max} \le \sqrt{1+\epsilon}$. Thus

$$\begin{aligned} d_{TV}(S, S_{ISO}) &\le 2\left(1 - \left(\frac{1-\epsilon}{1+\epsilon}\right)^{3n/2}\right) \\ &\le 2\left(1 - (1-\epsilon)^{3n}\right) \\ &\le 6n\epsilon. \end{aligned}$$

☐

6 The local and global maxima of the 3rd moment of the standard simplex and the isotropic simplex

In this section we study the structure of the set of local maxima of the third moment as a function of the direction (which happens to be essentially $u \mapsto \sum u_i^3$ as discussed in Section 3). This is not necessary for our algorithmic result, however it gives insight into the geometry of the third moment (the location of local maxima/minima and stationary points) and suggests that more direct optimization algorithms like gradient descent and Newton's method will also work, although we will not prove that.

Theorem 6. *Let $K \subseteq \mathbb{R}^n$ be an isotropic simplex. Let X be random in K. Let $V = \{x_i\}_{i=1}^{n+1} \subseteq \mathbb{R}^n$ be the set of normalized vertices of K. Then V is a complete set of local maxima and a complete set of global maxima of $F : S^{n-1} \to \mathbb{R}$ given by $F(u) = \mathbb{E}((u \cdot X)^3)$.*

Proof idea: Embed the simplex in $\mathbb{R}^{n+1}$. Show that the third moment is proportional to the complete homogeneous symmetric polynomial of degree 3, which for the relevant directions is proportional to the sum of cubes. To conclude, use first and second order optimality conditions to characterize the set of local maxima.

Proof. Consider the standard simplex

$$\Delta^n = \operatorname{conv}\{e_1, \dots, e_{n+1}\} \subseteq \mathbb{R}^{n+1}$$

and identify it with V via a linear map $A : \mathbb{R}^{n+1} \to \mathbb{R}^n$ so that $A(\Delta^n) = V$. Let Y be random in Δ^n. Consider $G : S^n \to \mathbb{R}$ given by $G(v) = m_3(v) = \mathbb{E}((v \cdot Y)^3)$. Let $U = \{v \in \mathbb{R}^{n+1} : v \cdot \mathbb{1} = 0, \|v\| = 1\}$ be the equivalent feasible set for the embedded problem. We have $G(v) = cF(Av)$ for any $v \in U$ and some constant $c > 0$ independent of v. To get the theorem, it is enough to show that the local maxima of G in U are precisely the normalized versions of the projections of the canonical vectors onto the hyperplane orthogonal to $\mathbb{1} = (1, \dots, 1)$. According to Section 3, for $v \in U$ we have

$$G(v) \propto p_3(v).$$

Using a more convenient but equivalent constant, we want to enumerate the local maxima of the problem

$$\begin{aligned} \max\ & \frac{1}{3} p_3(v) \\ \text{s.t.}\quad & v \cdot v = 1 \\ & v \cdot \mathbb{1} = 0 \\ & v \in \mathbb{R}^{n+1}. \end{aligned} \tag{6}$$

The Lagrangian function is

$$L(v, \lambda_1, \lambda_2) = \frac{1}{3} \sum_i v_i^3 - \lambda_1 \sum_i v_i - \lambda_2 \frac{1}{2} \left(\left(\sum_i v_i^2 \right) - 1 \right).$$

The first order condition is $\nabla_v L = 0$, that is,

$$v_i^2 = \lambda_1 + \lambda_2 v_i \quad \text{for } i = 1, \dots, n+1. \tag{7}$$

Consider this system of equations on v for any fixed λ_1, λ_2. Let $f(x) = x^2$, $g(x) = \lambda_1 + \lambda_2 x$. The first order condition says $f(v_i) = g(v_i)$, where f is convex and g is affine. That is, the v_is can take at most two different values. As our optimization problem (6) is symmetric under permutation of the coordinates, we conclude that, after putting the coordinates of a point v in non-increasing order, if v is a local maximum of (6), then v must be of the form

$$v = (a, \dots, a, b, \dots, b),$$

where $a > 0 > b$ and there are exactly α as and β bs, for $\alpha, \beta \in \{1, \dots, n\}$.

We will now study the second order necessary condition (SONC) to eliminate from the list of candidates all vectors with $\alpha > 1$. It is easy to see that the surviving vectors are exactly the promised scaled projections of the canonical vectors. This vectors must all be local and global maxima: At least one of them must be a global maximum as we are maximizing a continuous function over a compact set and all of them have the same objective value so all of them are local and global maxima.

The SONC at v asks for the Hessian of the Lagrangian to be negative semidefinite when restricted to the tangent space to the constraint set at v [16, Section 11.5]. We compute the Hessian (recall that $v^{(2)}$ is the vector of the squared coordinates of v):

$$\nabla_v L = v^{(2)} - \lambda_1 \mathbb{1} - \lambda_2 v$$

$$\nabla_v^2 L = 2\,\mathrm{diag}(v) - \lambda_2 I$$

where $\mathrm{diag}(v)$ is the $(n+1)$-by-$(n+1)$ matrix having the entries of v in the diagonal and 0 elsewhere.

A vector in the tangent space is any $z \in \mathbb{R}^{n+1}$ such that $z \cdot \mathbb{1} = 0$, $v \cdot z = 0$, and definiteness of the Hessian is determined by the sign of $z^T \nabla_v^2 L z$ for any such z, where

$$z^T \nabla_v^2 L z = \sum_{i=1}^{n+1} z_i^2 (2v_i - \lambda_2).$$

Suppose v is a critical point with $\alpha \geq 2$. To see that such a v cannot be a local maximum, it is enough to show $2a > \lambda_2$, as in that case we can take $z = (1, -1, 0, \dots, 0)$ to make the second derivative of L positive in the direction z.

In terms of α, β, a, b, the constraints of (6) are $\alpha a + \beta b = 0$, $\alpha a^2 + \beta b^2 = 1$, and this implies $a = \sqrt{\frac{\beta}{\alpha(n+1)}}$, $b = -\sqrt{\frac{\alpha}{\beta(n+1)}}$. The inner product between the first order condition (7) and v implies $\lambda_2 = \sum v_i^3 = \alpha a^3 + \beta b^3$. It is convenient to consider the change of variable $\gamma = \alpha/(n+1)$, as now candidate critical points are parameterized by certain discrete values of γ in $(0,1)$. This gives $\beta = (1-\gamma)(n+1)$, $a = \sqrt{(1-\gamma)/(\gamma(n+1))}$ and

$$\begin{aligned}\lambda_2 &= (n+1)\left[\gamma\left(\frac{1-\gamma}{\gamma(n+1)}\right)^{3/2}\right.\\&\quad\left.-(1-\gamma)\left(\frac{\gamma}{(1-\gamma)(n+1)}\right)^{3/2}\right]\\&= \frac{1}{\sqrt{(n+1)\gamma(1-\gamma)}}\left[(1-\gamma)^2-\gamma^2\right]\\&= \frac{1}{\sqrt{(n+1)\gamma(1-\gamma)}}[1-2\gamma].\end{aligned}$$

This implies

$$\begin{aligned}2a-\lambda_2 &= \frac{1}{\sqrt{(n+1)\gamma(1-\gamma)}}[2(1-\gamma)-1+2\gamma]\\&= \frac{1}{\sqrt{(n+1)\gamma(1-\gamma)}}.\end{aligned}$$

In $(0,1)$, the function given by $\gamma \mapsto 2a-\lambda_2 = \frac{1}{\sqrt{(n+1)\gamma(1-\gamma)}}$ is convex and symmetric around $1/2$, where it attains its global minimum value, $2/\sqrt{n+1}$, which is positive. □

7 Acknowledgments

We thank Santosh Vempala for telling us about the polytope learning problem, the approach of using higher-order moments and for helpful discussions. We also thank Keith Ball, Mikhail Belkin, Adam Kalai, Aaditya Ramdas and James Voss for helpful discussions.

References

[1] R. Adamczak, A. E. Litvak, A. Pajor, and N. Tomczak-Jaegermann. Quantitative estimates of the convergence of the empirical covariance matrix in log-concave ensembles. *J. Amer. Math. Soc.*, 23(2):535–561, 2010.

[2] A. Anandkumar, D. Hsu, and S. M. Kakade. A method of moments for mixture models and hidden markov models. In *COLT*, 2012. http://arxiv.org/abs/1203.0683.

[3] S. Arora, R. Ge, A. Moitra, and S. Sachdeva. Provable ICA with unknown Gaussian noise, and implications for Gaussian mixtures and autoencoders. In *NIPS*, 2012. arXiv:1206.5349.

[4] P. Comon and C. Jutten, editors. *Handbook of Blind Source Separation*. Academic Press, 2010.

[5] A. M. Frieze, M. Jerrum, and R. Kannan. Learning linear transformations. In *FOCS*, pages 359–368, 1996.

[6] N. Goyal and L. Rademacher. Learning convex bodies is hard. In *COLT*, 2009.

[7] N. Gravin, J. Lasserre, D. V. Pasechnik, and S. Robins. The inverse moment problem for convex polytopes. *Discrete & Computational Geometry*, 48(3):596–621, 2012.

[8] A. Grundmann and H. M. Moeller. Invariant integration formulas for the n-simplex by combinatorial methods. *SIAM J. Numer. Anal.*, 15:282–290, 1978.

[9] A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

[10] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis.* Wiley, 2001.

[11] A. R. Klivans, R. O'Donnell, and R. A. Servedio. Learning geometric concepts via Gaussian surface area. In *FOCS*, pages 541–550, 2008.

[12] A. R. Klivans and A. A. Sherstov. A lower bound for agnostically learning disjunctions. In *COLT*, pages 409–423, 2007.

[13] A. R. Klivans and A. A. Sherstov. Cryptographic hardness for learning intersections of halfspaces. *J. Comput. Syst. Sci.*, 75(1):2–12, 2009.

[14] S. Kwek and L. Pitt. PAC learning intersections of halfspaces with membership queries. *Algorithmica*, 22(1/2):53–75, 1998.

[15] J. B. Lasserre and K. E. Avrachenkov. The multi-dimensional version of $\int a^b x^p dx$. *American Math. Month.*, 108(2):151–154, 2001.

[16] D. G. Luenberger and Y. Ye. *Linear and nonlinear programming.* International Series in Operations Research & Management Science, 116. Springer, New York, third edition, 2008.

[17] P. Q. Nguyen and O. Regev. Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. *J. Cryptology*, 22(2):139–160, 2009.

[18] A. Packer. NP-hardness of largest contained and smallest containing simplices for V- and H-polytopes. *Discrete & Computational Geometry*, 28(3):349–377, 2002.

[19] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

[20] N. Srivastava and R. Vershynin. Covariance estimation for distributions with $2+\epsilon$ moments, 2011.

[21] S. Vempala. Learning convex concepts from Gaussian distributions with PCA. In *FOCS*, pages 124–130, 2010.

[22] S. Vempala. A random-sampling-based algorithm for learning intersections of halfspaces. *J. ACM*, 57(6):32, 2010.

[23] S. S. Vempala and Y. Xiao. Structure from local optima: Learning subspace juntas via higher order pca. *CoRR*, abs/1108.3329, 2011.